Machine Vision System for Urban Scenic Detection

Veejone Khoo
School of Engineering
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
veejone.here@gmail.com

Chandrasekharan Nataraj
School of Engineering
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
chandrasekharan@apu.edu.my

Lai Nai Shyan
School of Engineering
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
nai.shyan@apu.edu.my

***Abstract*— Optical flow such as the Gunnar-Farneback algorithm is used in traffic to find the displacement between two image frames. To find an anomaly in videos, many studies have used optical flow as a basis for surveillance analysis together with other technologies such as machine learning and deep learning. However, newer technologies require more data science expertise and more resources such as a labeled dataset. The proposed system utilizes the Gunnar-Farneback optical flow algorithm to detect displacement and explores the data analysis techniques by collecting the overall normalization values to flag an anomaly. The system is designed using personal computers, python, and MATLAB while other tools such as FFMPEG and Microsoft Excel are used for video segmentation and data representation. However, testing shows that there is more work needed to optimize the analysis technique for defining an anomaly in videos.**

Keywords*— *Video processing, Gunnar-Faranback algorithm, image optimization and analysis, optical flow algorithm

I. INTRODUCTION

Nowadays, Technology has been the most important thing and useful in our daily life, (Alsharif et al., 2021). Since Closed-Circuit Television (CCTV) has been invented, progress has been made to surveillance systems better in ways of how the technology is connected and how their video footage can be stored. Only quite recently, Neural Networks are used in this area to detect objects and people from video footage. (Gawande et al., 2020)

Think of a system that can make surveillance easier than sitting in a room looking through video footage 24/7. With the help of deep learning, these videos can be analysed and simplified to form a summary video or a reel of highlights of the day.

To elaborate further of the use of (CCTV), CCTV is said to be useful 29% of the time for British Transport Police investigators to solve criminal cases. In the study, only 65% of the time during the period of study where video footage is available to them. (Ashby, 2017) CCTV in this case is used to aid investigations after the fact that the crime has happened. If CCTV can be used to catch a criminal during the act, the usefulness of CCTV can probably increase significantly.

Another study done by Queen Denise P (Queen Denise P. Cuevas et al., 2016). Cuevas and their team in 'Effectiveness of CCTV cameras Installation in Crime Prevention' discussed that many establishments have devices deployed but only a few of them are monitored 24/7. This suggest that most of the CCTVs are not used for real-time detection but for aiding officers in crime investigation. The study also found that CCTVs are mostly used to deter violators and they are proved to be very effective in that area. As effective as it may be, CCTV as a true real-time monitoring device can be improved. This study may be outdated but the discussion is still relevant to the use of CCTV device and their security operators currently in year 2021. (Queen Denise P. Cuevas et al., 2016)

In an experiment of letting trained and non-trained participants in predicting incidents during a CCTV task shows that non-trained participants made more false positives than trained participants. Note that in the experiment, there were single screen CCTV task and also four screens CCTV task in which the task increased the difficulty in detecting behaviours. The study also suggests that using automation in assisting in this type of task is challenging across large number of screens. All the more reason to conduct the research in this area. (Stainer et al., 2021)

Thus, this research will be exploring the monitoring aspect of CCTV device and the end product of this system research could be called as an A.I. CCTV Operators or an A.I assistant for the CCTV security operators.

From observation, before the artificial intelligence technological developments, surveillance camera has a low percentage of usefulness for investigators and most of the time they are utilized only after an incident has occurred. If these cameras can be used to identify incidents instantly, the problem could be resolved. On the other hand, recording surveillance footage piles up enormous amount of raw video data. Summarising the data with interesting points or turning them into highlight clips can reduce the amount of data to be archived.

As introduced, the Artificial Intelligence and Deep Learning technology has improved surveillance effectively by using them to identify attributes of people within the computer vision. However, there are gaps that are needed to improve as multiple surveillance cameras need to work well together to identify situations as suggested in one of the studies in this area. (Sreenu & Saleem Durai, 2019) Another part of the problem is when there is a crowd in the vision, how it is harder to assess suspicious behaviour in a crowd. Crowd analysis in surveillance still remains to be an open issue due to its complexity and reliability when determining behaviour of a group. (Luque Sánchez et al., 2020).

The aim of this project is to develop a Machine Vision system for urban scenic detection. Detecting group behaviour using crowd analysis proved to be a challenging and emerging topic as suggested in a study. There are challenging elements

such as time complexity, bad weather conditions, real world dynamics, occlusions, and overlapping of objects.

Next, it is said that there is lack of commercially available solutions for crowd behaviour analysis. (Luque Sánchez et al., 2020) Hence, this project will tackle the area by implementing a software design or prototype that looks like a commercially available product. Its purpose will be highlighting unique video footage from footage that contains no useful information; essentially clipping highlights from a video. Finally, elements such as the false positive rates should be assessed as they determine the reliability in video analytic systems.

II. System Implementation

A. Data set

For focusing on an urban setting, a custom dataset will be used, this recording will take place in a small business shop and the CCTV footage will be angled approximately 45 degrees downward Zhang et al., (2018).

B. Video Segmentation and Scaling

To avoid video loading errors, a long video needs to be cut into shorts. In this system, the 20-minute-long video is cut into 3-second short clips and named accordingly. The current method of cutting the video is still done manually using FFMPEG tool. Rescaling video is also important so to lessen the computation power needed when running the optical flow program. By using the cv2 library for python, the video can be resized by manipulating the frame shape and writing it into a new file. In this project, the video is resized to 35 percent of its original size.

C. Optical Flow Method Implementation

Optical flow method uses pixel intensity to detect pixel direction changes by finding the displacement and representing the changes in Hue-Saturation-Value (HSV) from one frame to another (GeeksforGeeks, 2021). The Gunnar-Farneback Optical Flow computes the magnitude and direction in flow vectors such as the change of x over time and the change of y over time. The direction flow can be visualized by the hue, the distance can be visualized by the value of HSV (Farneb, 2003).

Gunnar-Farneback works by taking two mostly identical frames and running an algorithm to find the displacement between the two. The algorithm is based on polynomial expansion and uses it to “approximate neighborhood” of the pixel with polynomial .(Farneb, 2003). Then with the polynomial approximation, a displacement estimation can be made by solving for translation. This translation is the displacement estimation. The assumption made using this method is that the two frames are mostly identical while making sure the error is small enough to be useful.

D. Programming

The Optical Flow Farneback function needs input from the previous grayscale image and the second grayscale image to make displacement estimation. The pyr_scale is the pyramid-level scale for the down-sampling. The ‘levels’ makes sure that computation is done at multiple levels of resolution. Increasing this number will allow the algorithm to run for larger displacements between frames which would also mean computation increases.

Winsize controls the average window size for fast motion detection. The ‘iterations’ runs the search loop to find key points in each pyramid level. Poly_n is the size of the pixel neighborhood. Poly_sigma is the standard deviation of gaussian, making derivatives smoother. Flow is the computed flow image. Finally, flags are used to select the type of approximation or filter.

E. Array Normalization Values

After calculating the optical flow, the displacement can be converted into polar coordinate values using cv2.cartToPolar. These values are now kept as an array in ‘mag’ and the values can be normalized using the norm_mag code.

F. Representation of the flow

Fig.1 shows the optical flow representation arrow and HSV format for the image data set. Depending on the previous and the next image frame, the dots and lines representation will show the arrows pointing from the previous point to the next point in the image frame. Using the HSV format, however, shows the flow changes through colors. In this case, the detected changes are around the person but the body of the person remains because the algorithm cannot distinguish the changes with the same color. As a result, only the head and the corners of the person have been detected.

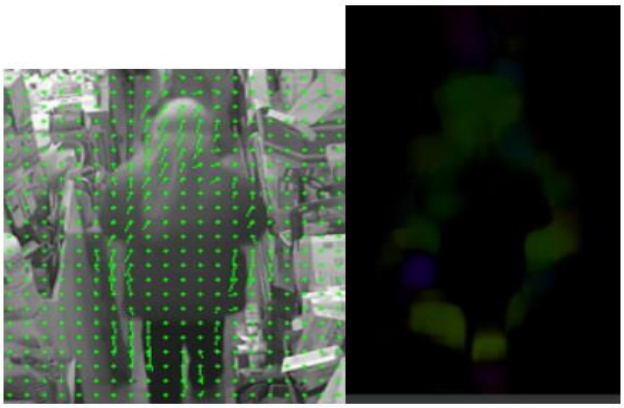

Fig. 1. Optical Flow Representation Arrow and HSV format

The block diagram of the system is shown in Fig.2 is the algorithm to process the information and then validate the results of the system.

- Input a video file into the system
- Resize the video
- Segment the video into 3-second clips
- Calculate the optical flow of a clip
- Convert the optical flow values into polar coordinate values
- Normalize the calculated values
- Determine if the values are greater than the normal values
    - Flag anomaly
- Else
    - Proceed
- Loop to the beginning until no more clips
- End

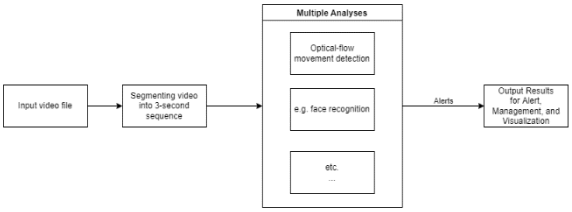

Fig. 2. Block Diagram of the system

III. SIMULATION RESULTS

In Fig 3., there are 5 people in it and the shop is operating as usual. The major movement in this scene can be observed on the right of the picture as the dots are pointing arrows to which the person is moving towards to. In the clip, people are behaving normally and moving as they normally would.

Fig. 3. Scene 1: A shop operating normally with 5 people in it

In Fig 4., it can be seen a person is swinging a weapon on the top left of the picture. Notice the direction arrows are slightly longer which indicates that there are large displacements between the two frames. In the clip, the person swings the weapon back and forth for the whole 3 seconds. This scenario is intentional created to show an example if violence were to happen in the scene.

Fig. 4. Scene 1: A person swinging a weapon

Fig. 5 illustrated the analysis of collected data using optical flow every 3 seconds. The data are normalized so there are only values between 0 and 1. The data are also categorized as green, yellow, and red. The green represents values greater than 0.5 and less than 0.7. The yellow are values between 0.7 and 0.9. Finally, the values greater than 0.9 are represented as the color red. Using this categorization technique can help in visualizing the data and seeing the patterns that the two distinct scenes produce. This happens because the flow in the clip is discontinuous.

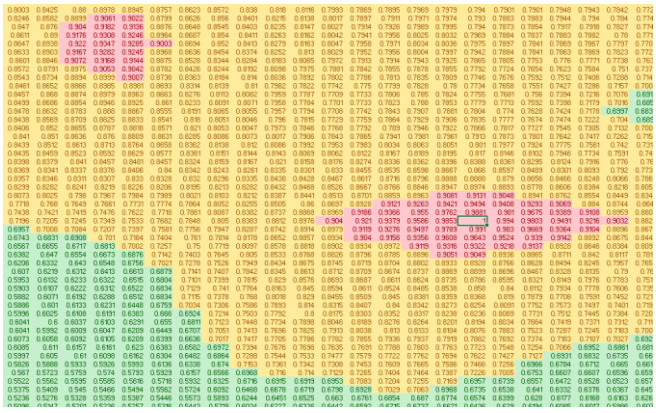

Fig. 5. An example of the data values categorized by color

Fig.6 is the scene captured and analysed by the optical flow algorithm. It is a normal scene with 5 people in it. The data shows the values having discontinuous values.

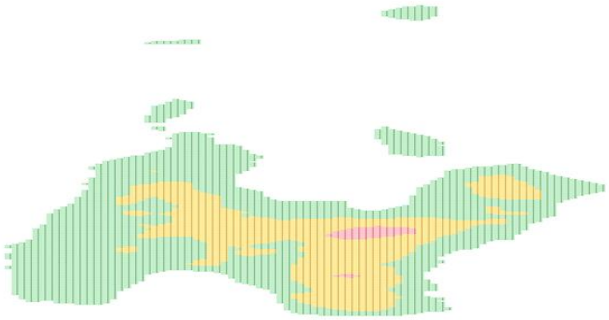

Fig. 6. Data Collected for Scene 1

Scene two illustrated in Fig.7 is intentionally made to capture the abnormal behavior of the humans. It is noticed that the swinging of the weapon creates a much smoother representation than in scene 1. The pattern of this scene can be clearly distinguished from scene 1. In general, this could mean that the flow of a typical day in the shop can be differentiated from abnormal ones.

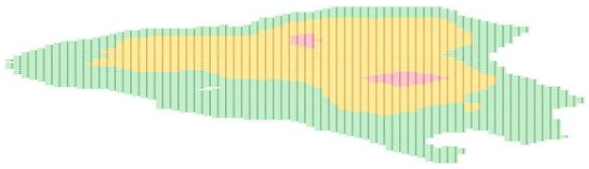

Fig. 7. Data collected for Scene 2

The validation for the anomaly in this work is based on the number count for the anomaly video clip in scene 2. A person swinging a weapon in which the person swings their weapon. The method is written so that only the values that are over 0.8 are counted. This would mean that the system is checking the overall scene for large displacement. This defines the anomaly as having large changes in the scene over the 3 seconds. For example, having five people moving around in the video will

make the calculated displacement high. Such as in scene 2, the system counts the number of values that are over 0.8 and found it to be about 600. Since this is the only anomaly video this project has, anything over 600 will be flagged as an anomaly.

IV. PROJECT FINDINGS

A. *Data analysis*

In this project, the 20-minute video is cut into 393 clips and each clip consists of 3 seconds of the video. These clips are then tested for anomaly using the system algorithm as shown in Fig 8.

133	count: 1610
134	Anomaly, for values of magnitude greater than 0.8, count is over 600
135	output039_rescaled done
136	output040_rescaled
137	count: 1253
138	Anomaly, for values of magnitude greater than 0.8, count is over 600
139	output040_rescaled done
140	count: 191
141	No anomaly
142	output041_rescaled done
143	count: 477
144	No anomaly
145	output042_rescaled done
146	output043_rescaled
147	count: 918
148	Anomaly, for values of magnitude greater than 0.8, count is over 600
149	output043_rescaled done
150	output044_rescaled
151	count: 889
152	Anomaly, for values of magnitude greater than 0.8, count is over 600
153	output044_rescaled done

Fig. 8. Testing for anomaly on 20 minutes of video data

After testing for anomaly Fig 9. shows the count of every clip. Clip 169 shows a count value of about 3300 which is the highest recorded. In the clip 169, it shows that there are major movement in four corners of the video which resulted in a tremendously scaled magnitude count.

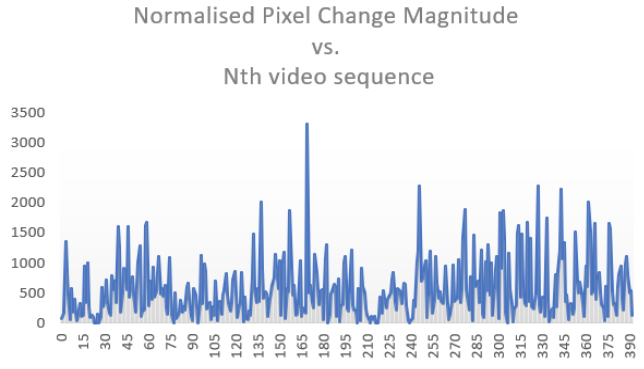

Fig. 9. Every clip and their count value in a graph

B. *Performance and Efficiency*

Testing video 152 shown in Fig 10**Error! Reference source not found.**, 11 and 12 gives an anomaly when it should not. This is due to the amount of movement occurred in a short period of time. This is expected because the design of the system did not calculate anomaly in the change in speed but only the change in magnitude of the pixels in a short time. The definition of anomaly should be optimised further in this system to ensure the reliability of the system as the only anomaly defined here is a person swinging an object. Otherwise, to flag an anomaly, a case by case can be done by comparing the number of people in a scene. Such that calculating an anomaly is based on 1 person setting, 2-person, 3-person, and so on. Then compare video in a testing environment of 1 person, 2-person, 3-person, and so on. If only there is a way to predict an anomaly based on all the normalized data collected instead of a rule-based system that this system has implemented.

Fig. 10. Video 152 shows 3 employees one the right and 1 customer on the left

```
output152_rescaled.mp4
count: 1014
Anomaly, for values of magnitude greater than 0.8, count is over 600
```

Fig. 11. Video 152 has 1014 counts of values over 0.8

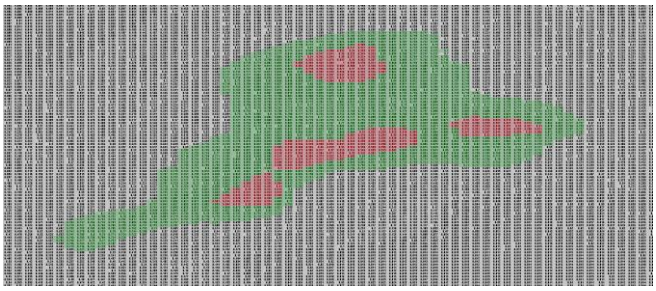

Fig. 12. Data representation for Video 152

Due to camera angle issue, tracking objects that are closer to the camera interferes with the algorithm used. Reliability can be increased if the system can categorize the depth of the scene. For example, Fig 13, a possible solution to this is to separate the frame into three parts, Close, Far, and Furthest. Segmenting it to make sure that the pixel magnitude values are weighted less than those that are further away from the camera. This solves the problem of magnified magnitude when objects are close to the camera.

Fig. 13. Magnitude is scaled when objects are closer to camera

C. *Performance and Efficiency*

The system can be improved in its performance by scaling down the resolution of the video. As shown in Table I, the average frames per second increased from 2fps to 15 fps when the resolution scaled down to 30% of its original resolution. This is because the number of pixels in the video is greatly reduced when the video is scaled down. Dense optical flow method such as the Gunnar-Farneback algorithm used calculates the displacement for every pixel in the frame. Scaling down the resolution causes the algorithm to process fewer pixels between frames which increases the overall performance of the system.

TABLE I. PERFORMANCE INCREASES WHEN RESOLUTION DROPS

Resolution	Average FPS
1920x1080	2
672x378	15

D. *Repeatability*

Fig 14. shows a person walking away from the camera and there are 1037 count of values over 0.8. Fig 15. shows person walking away and then towards the camera. There are 1680 count of values over 0.8. This means that a person walking towards the camera gives about 600 counts of values over 0.8. If there were any improvements to be made for this system, it would be scaling the depth of the camera using this piece of information.

Fig. 14. Person walking away from camera once

Conclusion

An algorithm has been implemented to detect suspicious behavior through anomaly detection via the Gunnar-Farneback optical flow method. The suspicious behavior is however defined as a lot of movement displacement over the 3-second analysis. Anomalous footage has been highlighted through the testing of 393 clips which is worth 20 minutes in total. This is done by calculating the magnitude of one anomalous clip such that a person is swinging their weapon and using the calculated magnitude to define it as the maximum magnitude for an anomaly. However, the method for validating anomalies can be further refined. The reliability of the system has been evaluated and further improvements can be made. The limitation of this system is the validation method. When an anomaly is poorly defined, the system flag normal situations as an anomaly. However, there are some consistent results such that when there are a lot of people and movement in the scene, the system does pick up a great amount of magnitude.

References

Ashby, M. P. J. (2017). The Value of CCTV Surveillance Cameras as an Investigative Tool: An Empirical Analysis. European Journal on Criminal Policy and Research, 23(3). https://doi.org/10.1007/s10610-017-9341-6.

Farneb, G. (2003). Two-Frame Motion Estimation Based on. Lecture Notes in Computer Science, 2749(1), 363–370.

Gawande, U., Hajari, K., & Golhar, Y. (2020). Pedestrian Detection and Tracking in Video Surveillance System: Issues, Comprehensive Review, and Challenges. In Recent Trends in Computational Intelligence. IntechOpen. https://doi.org/10.5772/intechopen.90810.

GeeksforGeeks. (2021). OpenCV – The Gunnar-Farneback optical flow. https://www.geeksforgeeks.org/opencv-the-gunnar-farneback-optical-flow/

Luque Sánchez, F., Hupont, I., Tabik, S., & Herrera, F. (2020). Revisiting crowd behaviour analysis through deep learning: Taxonomy, anomaly detection, crowd emotions, datasets, opportunities and prospects. Information Fusion, 64, 318–335. https://doi.org/10.1016/j.inffus.2020.07.008.

Queen Denise P. Cuevas, John Carlo P. Corachea, Ederlina B. Escabel, & Merwina Lou A. Bautista. (2016). Effectiveness of CCTV Cameras Installation in Crime Prevention. College of Criminology Research Journal, 7, 39–47.

Sreenu, G., & Saleem Durai, M. A. (2019). Intelligent video surveillance: a review through deep learning techniques for crowd analysis. In Journal of Big Data (Vol. 6, Issue 1). SpringerOpen. https://doi.org/10.1186/s40537-019-0212-5.

Stainer, M. J., Raj, P. v., Aitken, B. M., Bandarian-Balooch, S., & Boschen, M. J. (2021). Decision-making in single and multiple-screen CCTV surveillance. Applied Ergonomics, 93. https://doi.org/10.1016/j.apergo.2021.103383.

Zhang, X., Yu, Q., & Yu, H. (2018). Physics inspired methods for crowd video surveillance and analysis: A survey. IEEE Access, 6, 66816–66830. https://doi.org/10.1109/ACCESS.2018.2878733.

Alsharif, R. K., Nataraj, C., & Yusop, R. B. (2021). Machine vision analysis of harvested forestry sites using high resolution UAV data. *Journal of Applied Technology and Innovation*, *5*(1), 70-74.